In the Claims:

1. (Currently amended) A method for merging a plurality of virtual connections to form a merged virtual connection, comprising:

buffering ~~cells~~ data of each of the plurality of virtual connections into a corresponding one of a plurality of ~~cell~~ buffers, wherein each of the plurality of virtual connections is identified by an ~~virtual connection~~ identifier;

queuing the identity of a virtual connection in a queue when ~~cells~~ data that constitute a complete packet are buffered in a corresponding ~~cell~~ buffer;

obtaining prioritization information for the merged virtual connection; and

generating a ~~cell~~ data stream for the merged virtual connection based on the prioritization information and virtual connection identities stored in the queue, wherein the merged virtual connection is identified by a merged ~~virtual connection~~ identifier, wherein each ~~cell~~ unit of data in the ~~cell~~ data stream includes the merged ~~virtual connection~~ identifier.

2. (Currently amended) The method of claim 1, wherein generating the ~~cell~~ data stream for the merged virtual connection further comprises dequeuing ~~cells~~ data from the plurality of buffers to produce the ~~cell~~ data stream, wherein dequeuing of the ~~cells~~ data is based on the prioritization information.

3. (Original) The method of claim 1, wherein each of the plurality of virtual connections is included in a class of a plurality of classes, wherein the queue includes a plurality of queues, wherein each of the plurality of queues corresponds to a class of the plurality of classes, wherein queuing the identity of a virtual connection further comprises queuing the identity of the virtual connection into a corresponding one of the plurality of queues based on class of the virtual connection, wherein the prioritization information includes class prioritization information.

4. (Previously Presented) The method of claim 3, wherein each of the plurality of queues is a linked list, wherein queuing the identity of a virtual connection further comprises

appending the identity of the virtual connection to a tail of a corresponding one of the linked lists based on class of the virtual connection.

5. (Original) The method of claim 3, wherein the prioritization information allocates available bandwidth on the merged virtual connection based on class.

6. (Original) The method of claim 3, wherein obtaining prioritization information further comprises referencing a prioritization table that stores an accessing sequence for the plurality of queues.

7. (Currently amended) The method of claim 1, wherein generating the ~~cell~~ data stream further comprises generating the ~~cell~~ data stream such that ~~cells~~ data corresponding to different packets that are combined to produce the merged virtual connection are not intermingled.

8. (Currently amended) The method of claim 1, wherein determining that ~~cells~~ data that constitute a complete packet are buffered further comprises detecting an end of message indication that indicates a final ~~cell~~ unit of data for the complete packet.

9. (Currently amended) The method of claim 1 further comprises generating a ~~cell~~ stream for a port by combining the ~~cell~~ data stream for the merged virtual connection with a ~~cell~~ data stream corresponding to an additional virtual connection, wherein the ~~virtual connection~~ identifier corresponding to the additional virtual connection is different than the merged ~~virtual connection~~ identifier.

10. (Currently amended) A virtual connection merging system, comprising:

a plurality of buffers, wherein each buffer of the plurality of buffers corresponds to a virtual connection of a plurality of virtual connections, wherein each of the plurality of virtual connections has a unique ~~virtual connection~~ identifier, wherein ~~cells~~ data received for each of the plurality of virtual connections are buffered in corresponding buffers of the plurality of buffers;

a queuing block that stores identities of virtual connections that have complete packets buffered in the plurality of buffers, wherein a complete packet includes a plurality of ~~cells~~ units of data;

a prioritization block that stores prioritization information; and

a merging system controller operably coupled to the plurality of buffers, the queuing block, and the prioritization block, wherein the merging system controller dequeues ~~cells~~ data from the plurality of buffers based on the prioritization information and the identities stored in the queuing block to produce a ~~cell~~ data stream corresponding to a merged virtual connection, wherein ~~cells~~ data are dequeued from the plurality of buffers by the merging system controller such that ~~cells~~ data from a packet are included in a continuous portion of the ~~cell~~ data stream for the merged virtual connection, wherein ~~cells~~ data included in the ~~cell~~ data stream for the merged virtual connection include an ~~virtual connection~~ identifier corresponding to the merged virtual connection.

11. (Original) The virtual connection merging system of claim 10, wherein the queuing block further comprises a plurality of queues corresponding to a plurality of classes, wherein each of virtual connections corresponds to one of the plurality of classes, wherein identities of virtual connections for which complete packets are buffered are stored in corresponding queues of the plurality of queues based on class.

12. (Currently amended) The virtual connection merging system of claim 11, wherein the plurality of queues further comprises a plurality of linked lists, wherein when ~~cells~~ data that constitute a complete packet are buffered in one of the plurality of buffers, identity of a particular virtual connection to which the complete packet corresponds is added to a tail of the linked list for a class to which the particular virtual connection corresponds.

13. (Currently amended) The virtual connection merging system of claim 11, wherein the prioritization information stored in the prioritization block prioritizes inclusion of packets in the ~~cell~~ data stream for the merged virtual connection based on class.

14. (Currently amended) The virtual connection merging system of claim 13, wherein the prioritization information causes transitions between classes for dequeuing based on at least one of: a number of packets for a particular class included in the ~~cell~~ data stream for the merged virtual connection and a number of ~~cells~~ units of data for a particular class included in the ~~cell~~ data stream for the merged virtual connection.

15. (Original) The virtual connection merging system of claim 10, wherein the virtual connection merging system is included in the ingress portion of a communication switch.

16. (Original) The virtual connection merging system of claim 10, wherein the virtual connection merging system is included in the egress portion of a communication switch.

17. (Currently amended) A virtual connection merging processor, comprising:

a processing module; and

memory operably coupled to the processing module, wherein the memory stores operating instructions that, when executed by the processing module, cause the processing module to perform the functions of:

buffering ~~cells~~ data of each of a plurality of virtual connections into a corresponding one of a plurality of ~~cell~~ buffers, wherein each of the plurality of virtual connections is identified by a ~~virtual connection~~ identifier;

queuing the identity of a virtual connection in a queue when ~~cells~~ that constitute a complete packet are buffered in a corresponding ~~cell~~ buffer;

obtaining prioritization information for the merged virtual connection; and

generating a ~~cell~~ data stream for a merged virtual connection based on the prioritization information and virtual connection identities stored in the queue, wherein the merged virtual connection is identified by a merged ~~virtual connection~~ identifier, wherein each ~~cell~~ unit of data in the ~~cell~~ data stream includes the merged ~~virtual connection~~ identifier.

18. (Currently amended) The virtual connection merging processor of claim 17, wherein the memory further comprises operating instructions that, when executed by the processing module, cause the processing module to generate the ~~cell~~ data stream for the merged virtual connection by dequeuing data ~~cells~~ from the plurality of buffers to produce the ~~cell~~ data stream, wherein dequeuing of the ~~cells~~ data is based on the prioritization information.

19. (Original) The virtual connection merging processor of claim 17, wherein each of the plurality of virtual connections is included in a class of a plurality of classes, wherein the queue includes a plurality of queues, wherein each of the plurality of queues corresponds to a class of the plurality of classes, wherein the processing module queues the identity of a virtual connection by queuing the identity of the virtual connection into a corresponding one of the plurality of queues based on class of the virtual connection, wherein the prioritization information includes class prioritization information.

20. (Original) The virtual connection processor of claim 19, wherein each of the plurality of queues is a linked list, wherein the processing module queues the identity of a virtual connection by appending the identity of the virtual connection to a tail of a corresponding one of the linked lists based on class of the virtual connection.

21. (Original) The virtual connection processor of claim 19, wherein the prioritization information allocates available bandwidth on the merged virtual connection based on class.

22. (Original) The virtual connection processor of claim 19, wherein the processing module obtains prioritization information by referencing a prioritization table that stores an accessing sequence for the plurality of queues.

23. (Currently amended) The virtual connection processing module of claim 17, wherein the memory further comprises operating instructions such that the processing module performs the function of generating the ~~cell~~ data stream such that ~~cells~~ data corresponding to different packets that are combined to produce the merged virtual connection are not intermingled.

24. (Currently amended) The virtual connection processing module of claim 17, wherein the memory further comprises operating instructions such that the processing module determines that ~~cells~~ data that constitute a complete packet are buffered by detecting an end of message indication that indicates a final ~~cell~~ for the complete packet.

25. (Currently amended) A method for merging a plurality of virtual connections to form a merged virtual connection, comprising:

buffering ~~cells~~ data of each of the plurality of virtual connections into a corresponding one of a plurality of ~~cell~~ buffers, wherein each of the plurality of virtual connections is identified by a ~~virtual connection~~ identifier;

queuing the identity of a virtual connection in a queue when ~~cells~~ data that constitute a complete packet are buffered in a corresponding ~~cell~~ buffer;

obtaining prioritization information for the merged virtual connection; and

generating a ~~cell~~ data stream for the merged virtual connection based on the prioritization information and virtual connection identities stored in the queue, wherein the merged virtual connection is identified by a merged ~~virtual connection~~ identifier, wherein each ~~cell~~ unit of data in the ~~cell~~ data stream includes the merged ~~virtual connection~~ identifier, wherein dequeuing of ~~cells~~ data is performed in intervals, where different classes receive priority for different ones of the intervals.

26. (Currently amended) The method of claim 25, wherein generating the ~~cell~~ data stream for the merged virtual connection further comprises dequeuing ~~cells~~ data from the plurality of buffers to produce the ~~cell~~ data stream, wherein dequeuing of the ~~cells~~ data is based on the prioritization information.

27. (Previously presented) The method of claim 25, wherein each of the plurality of virtual connections is included in a class of a plurality of classes, wherein the queue includes a plurality of queues, wherein each of the plurality of queues corresponds to a class of the plurality of classes, wherein queuing the identity of a virtual connection further comprises queuing the identity of the virtual connection into a corresponding one of the plurality of queues based on class of the virtual connection, wherein the prioritization information includes class prioritization information.

28. (Previously presented) The method of claim 27, wherein each of the plurality of queues is a linked list, wherein queuing the identity of a virtual connection further comprises appending the identity of the virtual connection to a tail of a corresponding one of the linked lists based on class of the virtual connection.

29. (Previously presented) The method of claim 27, wherein the prioritization information allocates available bandwidth on the merged virtual connection based on class.

30. (Previously presented) The method of claim 27, wherein obtaining prioritization information further comprises referencing a prioritization table that stores an accessing sequence for the plurality of queues.

31. (Currently amended) The method of claim 25, wherein generating the ~~cell~~ data stream further comprises generating the ~~cell~~ data stream such that ~~cells~~ data corresponding to different packets that are combined to produce the merged virtual connection are not intermingled.

32. (Currently amended) The method of claim 25, wherein determining that ~~cells~~ data that constitute a complete packet are buffered further comprises detecting an end of message indication that indicates a final ~~cell~~ unit of data for the complete packet.

33. (Currently amended) The method of claim 25 further comprises generating a ~~cell~~ stream for a port by combining the ~~cell~~ data stream for the merged virtual connection with a ~~cell~~ data stream corresponding to an additional virtual connection, wherein the ~~virtual connection~~ identifier corresponding to the additional virtual connection is different than the merged ~~virtual connection~~ identifier.